

**IMPLEMENTASI *HORIZONTAL POD AUTOSCALER (HPA)* DALAM
SISTEM DETEKSI HOAKS BERBASIS ORKESTRASI *KUBERNETES***



**MUHAMMAD SABIL NUR RAIHAN
NIM. 2210511140**

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAKARTA
2026**

**IMPLEMENTASI *HORIZONTAL POD AUTOSCALER* (HPA) DALAM
SISTEM DETEKSI HOAKS BERBASIS ORKESTRASI *KUBERNETES***

**MUHAMMAD SABIL NUR RAIHAN
NIM. 2210511140**

SKRIPSI
**Sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer**

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
JAKARTA
2026**

LEMBAR PENGESAHAN

Judul : Implementasi Horizontal Pod Autoscaler (HPA) Dalam
Sistem Deteksi Hoaks Berbasis Orkestrasi Kubernetes
Nama : Muhammad Sabil Nur Raihan
NIM : 2210511140
Program Studi : S1 Informatika

Disetujui oleh:

Penguji 1:
Dr. Indra Permana Solihin, S.Kom, M.Kom.
NIP. 198308092021211001

Penguji 2:
Kharisma Wiati Gusti, M.T.
NIP. 199006192022032007

Pembimbing 1:
Noor Falih, S.Kom., M.T.
NIP. 198907152018031001

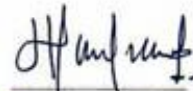
Pembimbing 2:
Muhammad Panji Muslim, S.Pd., M.Kom.
NIP. 199308312022031007

Diketahui oleh:

Koordinator Program Studi:
Dr. Ridwan Raafi'udin, S.Kom, M.Kom.
NIP. 198805022021211001

Dekan Fakultas Ilmu Komputer:
Prof. Dr. Ir. Supriyanto, M.Sc., IPM.
NIP. 197605082003121002

Tanggal Ujian Tugas Akhir:
13 Juli 2026



**PERSYARATAN MENGENAI SKRIPSI DAN SUMBER INFORMASI
SERTA PELIMPAAHAN HAK CIPTA**

Dengan ini saya menyatakan bahwa skripsi dengan judul "*IMPLEMENTASI HORIZONTAL POD AUTOSCALER (HPA) DALAM SISTEM DETEKSI HOAKS BERBASIS ORKESTRASI KUBERNETES*" adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Universitas Pembangunan Nasional "Veteran" Jakarta.

Jakarta, 16 Juli 2026



Muhammad Sabil Nur Raihan

NIM.2210511140

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK
KEPENTINGAN AKADEMIS**

Sebagai civitas akademika Universitas Pembangunan Nasional “Veteran” Jakarta,
saya yang bertanda tangan di bawah ini:

Nama	: Muhammad Sabil Nur Raihan
NIM	: 2210511140
Fakultas	: Fakultas Ilmu Komputer
Program Studi	: S1 Informatika

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan
kepada Universitas Pembangunan Nasional Veteran Jakarta Hak Bebas Royalti
Non eksklusif (*Non - exclusive Royalty Free Right*) atas skripsi saya yang berjudul:

**Implementasi *Horizontal Pod Autoscaler* (HPA) Dalam Sistem
Deteksi Hoaks Berbasis Orkestrasi *Kubernetes***

Dengan Hak Bebas Royalti ini Universitas Pembangunan Nasional “Veteran”
Jakarta berhak menyimpan, mengalih media atau memformatkan, mengelola
dalam bentuk pangkalan data (basis data), merawat dan mempublikasikan skripsi
saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemilik
Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 16 Juli 2026
Yang menyatakan,



**Muhammad Sabil Nur Raihan
NIM.2210511140**

IMPLEMENTASI *HORIZONTAL POD AUTOSCALER* (HPA) DALAM SISTEM DETEKSI HOAKS BERBASIS ORKESTRASI *KUBERNETES*

Muhammad Sabil Nur Raihan

ABSTRAK

Penelitian ini membahas implementasi *Horizontal Pod Autoscaler* (HPA) pada sistem deteksi hoaks berbasis orkestrasi *Kubernetes* yang mengintegrasikan pra-pengolahan citra, *Optical Character Recognition* (OCR), dan klasifikasi teks menggunakan *IndoBERT*. Sistem dirancang menggunakan arsitektur *service* dengan komunikasi asinkron untuk mengurangi *bottleneck* pada arsitektur monolitik. Implementasi dilakukan pada lingkungan lokal dan server, kemudian diuji menggunakan *Locust* melalui skenario *baseline*, *ramp-up*, *stress*, dan *soak*. Parameter pengujian meliputi *throughput*, P95 *latency*, penggunaan CPU, penggunaan memori, *error rate*, serta jumlah replica. Hasil pengujian menunjukkan bahwa arsitektur *service* menghasilkan *throughput* lebih tinggi dan P95 *latency* lebih rendah dibandingkan arsitektur monolitik. Pada skenario *stress*, *service* lokal dan *service* server mencapai 147 req/s dan 177 req/s, sedangkan monolitik lokal dan monolitik server hanya mencapai 2,67 req/s dan 0,689 req/s. HPA mampu meningkatkan replica hingga 25 pada *service* lokal dan 20 pada *service* server. Uji *Wilcoxon Signed-Rank* menghasilkan *p-value* 0,0078 pada *throughput* dan P95 *latency*, yang menunjukkan perbedaan kinerja signifikan. Dengan demikian, arsitektur *service* berbasis *Kubernetes* mampu meningkatkan kapasitas, responsivitas, dan skalabilitas sistem deteksi hoaks.

Kata Kunci: *Horizontal Pod Autoscaler*, *Kubernetes*, deteksi hoaks, OCR, *IndoBERT*, pengujian kinerja.

IMPLEMENTASI HORIZONTAL POD AUTOSCALER (HPA) DALAM SISTEM DETEKSI HOAKS BERBASIS ORKESTRASI KUBERNETES

Muhammad Sabil Nur Raihan

ABSTRACT

This study discusses the implementation of a Horizontal Pod Autoscaler (HPA) in a Kubernetes-orchestrated fake news detection system that integrates image preprocessing, Optical Character Recognition (OCR), and text classification using IndoBERT. The system was designed using a service-based architecture with asynchronous communication to reduce bottlenecks in the monolithic architecture. The implementation was deployed in local and server environments and evaluated using Locust through baseline, ramp-up, stress, and soak scenarios. The evaluation parameters included throughput, P95 latency, CPU usage, memory usage, error rate, and the number of replicas. The results show that the service-based architecture achieved higher throughput and lower P95 latency than the monolithic architecture. In the stress scenario, the local and server service configurations reached 147 req/s and 177 req/s, while the local and server monolithic configurations reached only 2.67 req/s and 0.689 req/s. The HPA increased the replicas to 25 in the local service configuration and 20 in the server service configuration. The Wilcoxon Signed-Rank Test produced a p-value of 0.0078 for throughput and P95 latency, indicating a significant performance difference. Therefore, the Kubernetes-based service architecture improved the capacity, responsiveness, and scalability of the fake news detection system.

Keywords: Horizontal Pod Autoscaler, Kubernetes, fake news detection, OCR, IndoBERT, performance testing.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “IMPLEMENTASI HORIZONTAL POD AUTOSCALER (HPA) DALAM SISTEM DETEKSI HOAKS BERBASIS ORKESTRASI KUBERNETES”. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jakarta.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, arahan, dan doa dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada:

1. Orang tua dan keluarga penulis yang selalu memberikan doa, dukungan moral, serta dukungan materi selama proses perkuliahan dan penyusunan skripsi ini.
2. Bapak Prof. Dr. Ir. Supriyanto, M.Sc., IPM. selaku Dekan Fakultas Ilmu Komputer Universitas Pembangunan Nasional “Veteran” Jakarta.
3. Bapak Dr. Ridwan Raafi’udin, S.Kom., M.Kom. selaku Koordinator Program Studi Informatika Program Sarjana Fakultas Ilmu Komputer Universitas Pembangunan Nasional “Veteran” Jakarta.
4. Bapak Dr. Noor Falih, S.Kom., M.T. selaku dosen pembimbing pertama yang telah meluangkan waktu, memberikan arahan, serta membimbing penulis dalam proses penyusunan skripsi ini.
5. Bapak Muhammad Panji Muslim, S.Pd., M.Kom. selaku dosen pembimbing kedua yang telah memberikan bimbingan, masukan, dan dukungan selama penyusunan skripsi ini.
6. Bapak Hamonangan Kinantan Prabu, M.T. selaku dosen pembimbing akademik yang telah memberikan arahan, informasi, dan dukungan selama masa perkuliahan.
7. Seluruh sahabat penulis yang tidak dapat disebutkan satu per satu, yang selalu memberikan dukungan, semangat, dan bantuan selama proses penyusunan skripsi ini.
8. Teman-teman seperjuangan yang telah menemani, mendukung, dan mewarnai perjalanan penulis selama masa perkuliahan hingga penyusunan skripsi.
9. Teman-teman dari Discord 3K, Dimare Ajh Cuy, NYPD, dan Sin D Cat yang telah memberikan bantuan, dukungan, serta semangat kepada penulis.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, penulis berharap pembaca memberikan kritik dan saran yang konstruktif.

Jakarta, 16 Juli 2026

Penulis

DAFTAR ISI

KATA PENGANTAR.....	viii
DAFTAR ISI	x
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL	xix
DAFTAR RUMUS	xx
DAFTAR LAMPIRAN	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan dan Manfaat Penelitian	5
1.5. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA	7
2.1. Kajian Teoritis.....	7
2.1.1. Karakteristik Beban Kerja Sistem Deteksi Hoaks	7
2.1.2. <i>Computational Overhead</i> pada OCR dan <i>Deep Learning</i>	7
2.1.3. Arsitektur Perangkat Lunak (<i>Software Architecture</i>)	8
2.1.3.1 Arsitektur Monolitik.....	10
2.1.3.2 Arsitektur Berbasis <i>Service</i>	11
2.1.3.3 Komunikasi Asinkron dalam Arsitektur Berbasis <i>Service</i> ..	12
2.1.4. Teknologi Kontainerisasi dan Orkestrasi	14
2.1.4.1 <i>Docker</i>	15
2.1.4.2 <i>Kubernetes</i>	16
2.1.5. <i>Horizontal Pod Autoscaler</i> (HPA)	18
2.1.6. <i>Performance Engineering</i> dan Metrik Kinerja	19
2.1.6.1 Pengukuran Efisiensi Sumber Daya (<i>Resource Efficiency</i>) .	20
2.1.6.2 Analisis P95 Latensi	21
2.1.6.3 <i>Throughput</i>	22
2.1.6.4 <i>Load Testing</i>	23
2.1.7. Arsitektur <i>Pipeline</i> Sistem Deteksi Hoaks	24
2.2. Penelitian Terdahulu	27
BAB III METODE PENELITIAN	30
3.1. Desain Penelitian	30

3.1.1.	Klarifikasi Penelitian (<i>Research Clarification</i>)	33
3.1.2.	Studi Deskriptif 1 (<i>Descriptive Study I</i>)	34
3.1.3.	Studi Preskriptif (<i>Prescriptive Study</i>)	34
3.1.4.	Studi Deskriptif 2 (<i>Descriptive Study II</i>).....	36
3.2.	Instrumen Penelitian	37
3.3.	Konfigurasi <i>Horizontal Pod Autoscaler</i>	38
3.4.	Rancangan Eksperimen Pengujian Kinerja	39
3.4.1.	Uji Validasi Parameter <i>Workload</i>	39
3.4.2.	Matriks Rancangan Eksperimen.....	42
3.5.	<i>Requirements Traceability Matrix</i>	43
3.6.	Alat dan Bahan Penelitian.....	46
3.6.1.	Perangkat Keras (<i>Hardware</i>).....	46
3.6.2.	Perangkat Lunak (<i>Software</i>).....	48
3.7.	Jadwal Penelitian.....	51
BAB IV HASIL DAN PEMBAHASAN.....		54
4.1.	Lingkungan Penelitian	54
4.2.	Skenario Pengujian	54
4.2.1.	<i>Endpoint</i> Pengujian.....	54
4.2.2.	Parameter Skenario Pengujian	55
4.2.3.	Hasil Implementasi Pemrosesan Asinkron	56
4.2.4.	Hasil Implementasi <i>Horizontal Pod Autoscaler</i>	58
4.3.	Pengujian <i>Baseline</i>	59
4.3.1.	Monolitik Lokal	60
4.3.1.1	<i>Throughput (Request Per Second)</i>	60
4.3.1.2	<i>Latency</i>	61
4.3.1.3	Penggunaan <i>CPU</i>	62
4.3.1.4	Penggunaan Memori.....	62
4.3.1.5	<i>Error Rate</i>	63
4.3.2.	<i>Service</i> Lokal	64
4.3.2.1	<i>Throughput (Request Per Second)</i>	64
4.3.2.2	<i>Latency</i>	65
4.3.2.3	Penggunaan <i>CPU</i> dan <i>HPA</i>	66

4.3.2.4	Penggunaan Memori.....	67
4.3.2.5	Error Rate	68
4.3.3.	Monolitik Server	69
4.3.3.1	Throughput (Request Per Second)	69
4.3.3.2	Latency.....	70
4.3.3.3	Penggunaan CPU	71
4.3.3.4	Penggunaan Memori.....	71
4.3.3.5	Error Rate	72
4.3.4.	Service Server.....	73
4.3.4.1	Throughput (Request Per Second)	73
4.3.4.2	Latency.....	74
4.3.4.3	Penggunaan CPU dan HPA	75
4.3.4.4	Penggunaan Memori.....	76
4.3.4.5	Error Rate	76
4.3.5.	Perbandingan Pengujian Baseline.....	77
4.4.	Pengujian Ramp-Up.....	79
4.4.1.	Monolitik Lokal	79
4.4.1.1	Throughput (Request Per Second)	79
4.4.1.2	Latency.....	80
4.4.1.3	Penggunaan CPU	81
4.4.1.4	Penggunaan Memori.....	82
4.4.1.5	Error Rate	82
4.4.2.	Service Lokal	83
4.4.2.1	Throughput (Request Per Second)	83
4.4.2.2	Latency.....	84
4.4.2.3	Penggunaan CPU dan Respons HPA	85
4.4.2.4	Penggunaan Memori dan Respons HPA.....	86
4.4.2.5	Error Rate	87
4.4.3.	Monolitik Server	88
4.4.3.1	Throughput (Request Per Second)	88
4.4.3.2	Latency.....	89
4.4.3.3	Penggunaan CPU	89

4.4.3.4	Penggunaan Memori.....	90
4.4.3.5	Error Rate	91
4.4.4.	Service Server.....	91
4.4.4.1	Throughput (Request Per Second)	91
4.4.4.2	Latency.....	92
4.4.4.3	Penggunaan CPU dan Respons HPA	93
4.4.4.4	Penggunaan Memori.....	94
4.4.4.5	Error Rate	95
4.4.5.	Perbandingan Pengujian Ramp-Up.....	96
4.5.	Pengujian Stress.....	97
4.5.1.	Monolitik Lokal	98
4.5.1.1	Throughput (Request Per Second)	98
4.5.1.2	Latency.....	99
4.5.1.3	Penggunaan CPU	100
4.5.1.4	Penggunaan Memori.....	101
4.5.1.5	Error Rate	102
4.5.2.	Service Lokal	102
4.5.2.1	Throughput (Request Per Second)	102
4.5.2.2	Latency.....	103
4.5.2.3	Penggunaan CPU dan Respons HPA	104
4.5.2.4	Penggunaan Memori dan HPA.....	105
4.5.2.5	Error Rate	106
4.5.3.	Monolitik Server	107
4.5.3.1	Throughput (Request Per Second)	107
4.5.3.2	Latency.....	108
4.5.3.3	Penggunaan CPU	109
4.5.3.4	Penggunaan Memori.....	110
4.5.3.5	Error Rate	110
4.5.4.	Service Server.....	111
4.5.4.1	Throughput (Request Per Second)	111
4.5.4.2	Latency.....	112
4.5.4.3	Penggunaan CPU dan HPA	113

4.5.4.4	Penggunaan Memori.....	114
4.5.4.5	Error Rate	115
4.5.5.	Perbandingan Pengujian Stress	116
4.6.	Pengujian Soak	118
4.6.1.	Monolitik Lokal	118
4.6.1.1	Throughput (Request Per Second)	118
4.6.1.2	Latency.....	119
4.6.1.3	Penggunaan CPU	121
4.6.1.4	Penggunaan Memori.....	121
4.6.1.5	Error Rate	122
4.6.2.	Service Lokal	122
4.6.2.1	Throughput (Request Per Second)	122
4.6.2.2	Latency.....	123
4.6.2.3	Penggunaan CPU dan HPA	124
4.6.2.4	Penggunaan Memori.....	125
4.6.2.5	Error Rate	126
4.6.3.	Monolitik Server	127
4.6.3.1	Throughput (Request Per Second)	127
4.6.3.2	Latency.....	128
4.6.3.3	Penggunaan CPU	128
4.6.3.4	Penggunaan Memori.....	129
4.6.3.5	Error Rate	130
4.6.4.	Service Server.....	130
4.6.4.1	Throughput (Request Per Second)	130
4.6.4.2	Latency.....	131
4.6.4.3	Penggunaan CPU dan HPA	132
4.6.4.4	Penggunaan Memori.....	133
4.6.4.5	Error Rate	134
4.6.5.	Perbandingan Pengujian Soak	135
4.7.	Analisis Respons HPA terhadap CPU, Jumlah Pod, dan Throughput 137	
4.8.	Pengujian Hipotesis Menggunakan Wilcoxon Signed-Rank Test..	139

4.8.1.	Perumusan Hipotesis	139
4.8.2.	Tahapan Perhitungan <i>Wilcoxon Signed-Rank Test</i>	140
4.8.3.	Perhitungan <i>Wilcoxon</i> pada Parameter <i>Throughput</i>	141
4.8.4.	Perhitungan <i>Wilcoxon</i> pada Parameter <i>P95 Latency</i>	143
4.8.5.	Perhitungan <i>P-Value Eksak</i>	144
4.8.6.	Ringkasan Hasil Uji <i>Wilcoxon Signed-Rank</i>	145
4.8.7.	Hasil Uji <i>Wilcoxon Signed-Rank</i>	146
4.9.	<i>Analisis Trade-off</i> Arsitektur Monolitik dan <i>Service</i>	146
4.10.	<i>Threats to Validity</i> (Keterbatasan Penelitian)	147
4.11.	Implementasi Antarmuka Web sebagai Pendukung Sistem.....	148
4.11.1.	Halaman Autentikasi Pengguna	149
4.11.2.	<i>Dashboard</i> dan Input Citra.....	150
4.11.3.	Tampilan Hasil Prediksi	151
4.11.4.	Riwayat Prediksi.....	152
4.11.5.	Konfigurasi <i>Domain</i> dan <i>Cloudflare</i>	152
4.11.6.	Kontribusi Implementasi Web	153
BAB V KESIMPULAN.....		154
5.1.	Kesimpulan	154
5.2.	Saran	155
DAFTAR PUSTAKA.....		157
DAFTAR LAMPIRAN		163

DAFTAR GAMBAR

Gambar 2.1.3. Arsitektur Monolitik vs <i>Microservices</i> (Sumber: Stelitano L, 2024)	9
Gambar 2.1.3.1 Arsitektur Monolitik (Sumber: Geeks For Geeks, 2025)	10
Gambar 2.1.3.2 Arsitektur Berbasis <i>Services</i> (Sumber: Harira Irawan et al., 2025)	11
Gambar 2.1.3.3 Komunikasi Asinkron (Sumber: Microsoft, 2026)	13
Gambar 2.1.4. <i>Kubernetes Cluster</i> (Sumber: Kubernetes, 2025)	15
Gambar 2.1.4.1 <i>Docker</i> (Sumber: Docker, 2025)	16
Gambar 2.1.4.2 <i>Kubernetes</i> (Sumber: Kubernetes, 2025)	17
Gambar 2.1.5 <i>Horizontal Pod Autoscaler</i> (HPA) (Sumber: Karthik, 2024) ...	18
Gambar 2.1.7 <i>Image Adaptive thresholding</i> (Sumber: Liang et al., 2023)	24
Gambar 2.1.7 <i>Image Deskewing</i> (Sumber: J & T, 2024)	25
Gambar 2.1.7 <i>Image Denoising</i> (Sumber: Vizcarra et al., 2024)	25
Gambar 3.1 <i>Design Research Methodology</i> (DRM)	32
Gambar 3.1.1 <i>Existing System Architecture Diagram</i>	33
Gambar 3.1.3 <i>Deployment Diagram</i> Arsitektur <i>Service</i> pada <i>Kubernetes</i>	36
Gambar 4.2.3. Alur Implementasi Pemrosesan Asinkron pada Arsitektur Berbasis <i>Service</i>	57
Gambar 4.2.4.1 Status <i>Horizontal Pod Autoscaler</i> pada <i>Namespace Hoax Detection</i>	58
Gambar 4.2.4.2 Konfigurasi Kebijakan <i>Scale-Up</i> dan <i>Scale-Down HPA</i>	59
Gambar 4.3.1.1 Hasil <i>Throughput RPS</i>	60
Gambar 4.3.1.2 Hasil <i>Latency</i>	61
Gambar 4.3.1.3 Hasil <i>CPU Usage</i>	62
Gambar 4.3.1.4 Hasil <i>Memory Usage</i>	63
Gambar 4.3.1.5 Hasil <i>Error Rate</i>	64
Gambar 4.3.2.1 Hasil <i>Throughput RPS</i>	65
Gambar 4.3.2.2 Hasil <i>Latency</i>	66
Gambar 4.3.2.3 <i>CPU Usage</i> dan <i>HPA</i>	66
Gambar 4.3.2.4 Hasil <i>Memory Usage</i>	68
Gambar 4.3.2.5 Hasil <i>Error Rate</i>	68
Gambar 4.3.3.1 Hasil <i>Throughput RPS</i>	69
Gambar 4.3.3.2 Hasil <i>Latency</i>	70
Gambar 4.3.3.3 Hasil <i>CPU Usage</i>	71
Gambar 4.3.3.4 Hasil <i>Memory Usage</i>	72
Gambar 4.3.3.5 Hasil <i>Error Rate</i>	72
Gambar 4.3.4.1 Hasil <i>Throughput RPS</i>	73
Gambar 4.3.4.2 Hasil <i>Latency</i>	74
Gambar 4.3.4.3 Hasil <i>CPU Usage</i> dan <i>HPA</i>	75
Gambar 4.3.4.4 Hasil <i>Memory Usage</i>	76
Gambar 4.3.4.5 Hasil <i>Error Rate</i>	77

Gambar 4.4.1.1 Hasil <i>Throughput RPS</i>	80
Gambar 4.4.1.2 Hasil <i>Latency</i>	81
Gambar 4.4.1.3 Hasil <i>CPU Usage</i>	82
Gambar 4.4.1.4 Hasil <i>Memory Usage</i>	82
Gambar 4.4.1.5 Hasil <i>Error Rate</i>	83
Gambar 4.4.2.1 Hasil <i>Throughput RPS</i>	84
Gambar 4.4.2.2 Hasil <i>Latency</i>	85
Gambar 4.4.2.3 <i>CPU Usage</i> dan <i>HPA</i>	86
Gambar 4.4.2.4 <i>Memory Usage</i> dan <i>HPA</i>	87
Gambar 4.4.2.5 Hasil <i>Error Rate</i>	87
Gambar 4.4.3.1 Hasil <i>Throughput RPS</i>	88
Gambar 4.4.3.2 Hasil <i>Latency</i>	89
Gambar 4.4.3.3 Hasil <i>CPU Usage</i>	90
Gambar 4.4.3.4 Hasil <i>Memory Usage</i>	90
Gambar 4.4.3.5 Hasil <i>Error Rate</i>	91
Gambar 4.4.4.1 Hasil <i>Throughput RPS</i>	92
Gambar 4.4.4.2 Hasil <i>Latency</i>	93
Gambar 4.4.4.3 Hasil <i>CPU Usage</i> dan <i>HPA</i>	94
Gambar 4.4.4.4 Hasil <i>Memory Usage</i>	95
Gambar 4.4.4.5 Hasil <i>Error Rate</i>	95
Gambar 4.5.1.1 Hasil <i>Throughput RPS</i>	98
Gambar 4.5.1.2 Hasil <i>Latency</i>	99
Gambar 4.5.1.3 Hasil <i>CPU Usage</i>	101
Gambar 4.5.1.4 Hasil <i>Memory Usage</i>	101
Gambar 4.5.1.5 Hasil <i>Error Rate</i>	102
Gambar 4.5.2.1 Hasil <i>Throughput RPS</i>	103
Gambar 4.5.2.2 Hasil <i>Latency</i>	104
Gambar 4.5.2.3 Hasil <i>CPU Usage</i> dan <i>HPA</i>	105
Gambar 4.5.2.4 Hasil <i>Memory Usage</i>	106
Gambar 4.5.2.5 Hasil <i>Error Rate</i>	107
Gambar 4.5.3.1 Hasil <i>Throughput RPS</i>	108
Gambar 4.5.3.2 Hasil <i>Latency</i>	109
Gambar 4.5.3.3 Hasil <i>CPU Usage</i>	109
Gambar 4.5.3.4 Hasil <i>Memory Usage</i>	110
Gambar 4.5.3.5 Hasil <i>Error Rate</i>	111
Gambar 4.5.4.1 Hasil <i>Throughput RPS</i>	112
Gambar 4.5.4.2 Hasil <i>Latency</i>	113
Gambar 4.5.4.3 Hasil <i>CPU Usage</i> dan <i>HPA</i>	114
Gambar 4.5.4.4 Hasil <i>Memory Usage</i>	115
Gambar 4.5.4.5 Hasil <i>Error Rate</i>	116
Gambar 4.6.1.1 Hasil <i>Throughput RPS</i>	119
Gambar 4.6.1.2 Hasil <i>Latency</i>	120

Gambar 4.6.1.3 Hasil <i>CPU Usage</i>	121
Gambar 4.6.1.4 Hasil <i>Memory Usage</i>	121
Gambar 4.6.1.5 Hasil <i>Error Rate</i>	122
Gambar 4.6.2.1 Hasil <i>Throughput RPS</i>	123
Gambar 4.6.2.2 Hasil <i>Latency</i>	124
Gambar 4.6.2.3 Hasil <i>CPU Usage</i> dan <i>HPA</i>	125
Gambar 4.6.2.4 Hasil <i>Memory Usage</i>	126
Gambar 4.6.2.5 Hasil <i>Error Rate</i>	126
Gambar 4.6.3.1 Hasil <i>Throughput RPS</i>	127
Gambar 4.6.3.2 Hasil <i>Latency</i>	128
Gambar 4.6.3.3 Hasil <i>CPU Usage</i>	129
Gambar 4.6.3.4 Hasil <i>Memory Usage</i>	129
Gambar 4.6.3.5 Hasil <i>Error Rate</i>	130
Gambar 4.6.4.1 Hasil <i>Throughput RPS</i>	131
Gambar 4.6.4.2 Hasil <i>Latency</i>	132
Gambar 4.6.4.3 Hasil <i>CPU Usage</i> dan <i>HPA</i>	133
Gambar 4.6.4.4 Hasil <i>Memory Usage</i>	134
Gambar 4.6.4.5 Hasil <i>Error Rate</i>	135
Gambar 4.11.1 Halaman Autentikasi Aplikasi Deteksi Hoaks	149
Gambar 4.11.2 <i>Dashboard</i> dan Fitur Unggah Citra	150
Gambar 4.11.2 Fitur Pengambilan Citra Menggunakan Kamera	151
Gambar 4.11.3 Tampilan Hasil Prediksi Sistem.....	151
Gambar 4.11.4 Tampilan Riwayat Prediksi.....	152
Gambar 4.11.4 Detail Riwayat Prediksi.....	152
Gambar 4.11.5 Tampilan <i>Cloudflare</i> Aktif	153

DAFTAR TABEL

Tabel 2.2 Penelitian Terdahulu	27
Tabel 3.4.1 Uji Validitas Parameter	39
Tabel 3.4.2 Matriks Rancangan Eksperimen.....	42
Tabel 3.5 <i>Requirements Traceability Matrix</i>	43
Tabel 3.6.1 Perangkat Keras	47
Tabel 3.6.2 Perangkat Lunak	48
Tabel 3.7 Jadwal Penelitian	51
Tabel 4.2.1 <i>Endpoint</i> Utama Pengujian.....	55
Tabel 4.2.2 Skenario Pengujian.....	56
Tabel 4.3.5 Hasil Perbandingan Pengujian <i>Baseline</i>	77
Tabel 4.4.5 Hasil Perbandingan Pengujian <i>Ramp - Up</i>	96
Tabel 4.5.5 Hasil Perbandingan Pengujian <i>Stress</i>	116
Tabel 4.6.5 Hasil Perbandingan Pengujian <i>Soak</i>	136
Tabel 4.7. Rekapitulasi <i>CPU</i> , Jumlah <i>Pod</i> , dan <i>Throughput</i> pada <i>Arsitektur Service</i>	137
Tabel 4.8.3 Perhitungan Selisih <i>Throughput</i>	141
Tabel 4.8.3 Urutan Peringkat Selisih <i>Throughput</i>	142
Tabel 4.8.4 Perhitungan Selisih <i>P95 Latency</i>	143
Tabel 4.8.4 Urutan Peringkat Selisih <i>P95 Latency</i>	144
Tabel 4.8.6 Hasil Uji <i>Wilcoxon Signed-Rank</i>	145

DAFTAR RUMUS

(2.1.5).....	18
(2.1.6.1).....	21
(2.1.6.2).....	21
(2.1.6.2).....	22
(2.1.6.3).....	22
(4.8.2).....	140
(4.8.2).....	140
(4.8.2).....	140
(4.8.2).....	140
(4.8.2).....	141
(4.8.3).....	141
(4.8.3).....	142
(4.8.3).....	142
(4.8.3).....	143
(4.8.3).....	143
(4.8.4).....	143
(4.8.4).....	143
(4.8.4).....	144
(4.8.4).....	144
(4.8.4).....	144
(4.8.4).....	144
(4.8.5).....	144
(4.8.5).....	145
(4.8.5).....	145
(4.8.5).....	145

DAFTAR LAMPIRAN

Lampiran 1.	Hasil Turnitin.....	163
--------------------	----------------------------	------------